UNIVERSITY OF ROCHESTER, ECE DEPARTMENT

ADVANCE CMOS VLSI

ASSIGNMENT -5

Submitted By CHINMAI

CONTENT

I. Part A……2

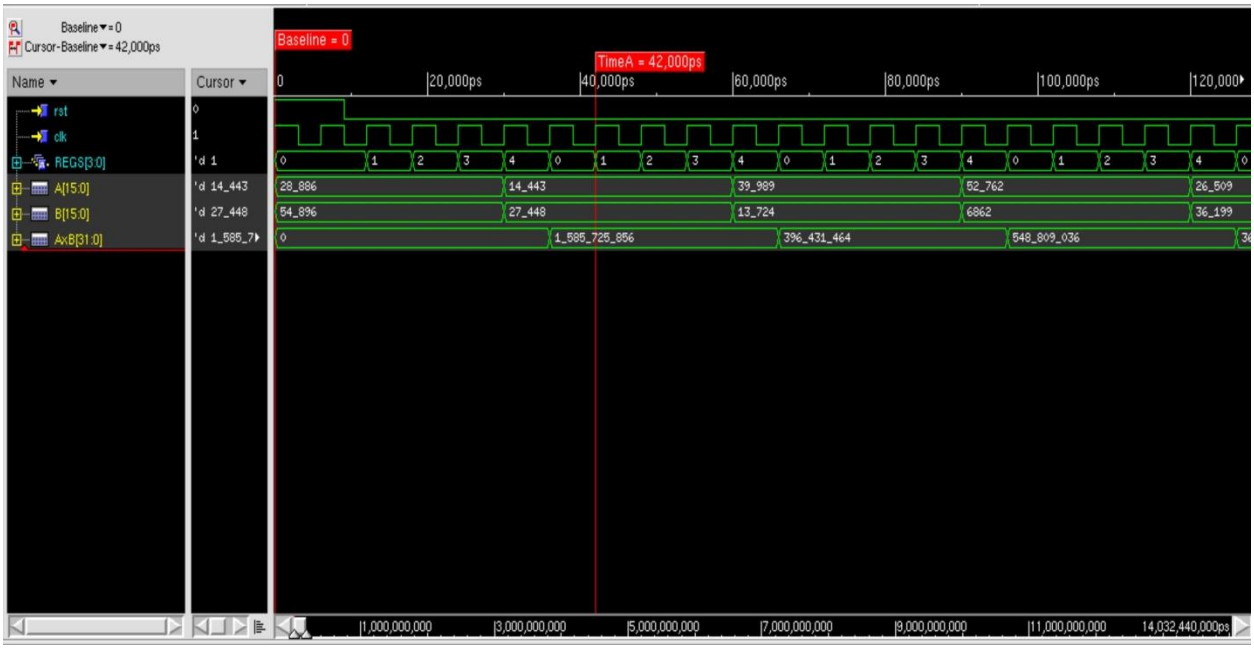
II. Part B……3

III. Part C……5

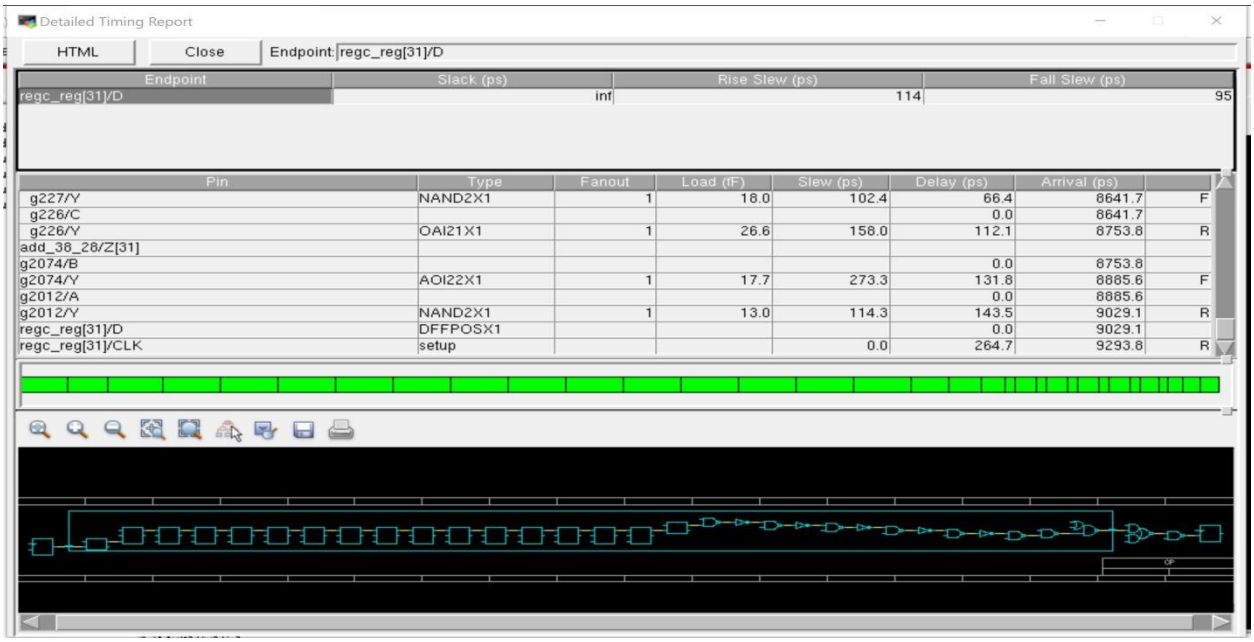
Part I

16X16 Multiplier

A(28,886) * B(54,896) = C (1,585,725,856) Time take to compute the result by this Multiplier is 33000ps



Worse Case Path Delay for the Multiplier measured is 264ps



Part II

DBNS Converter:

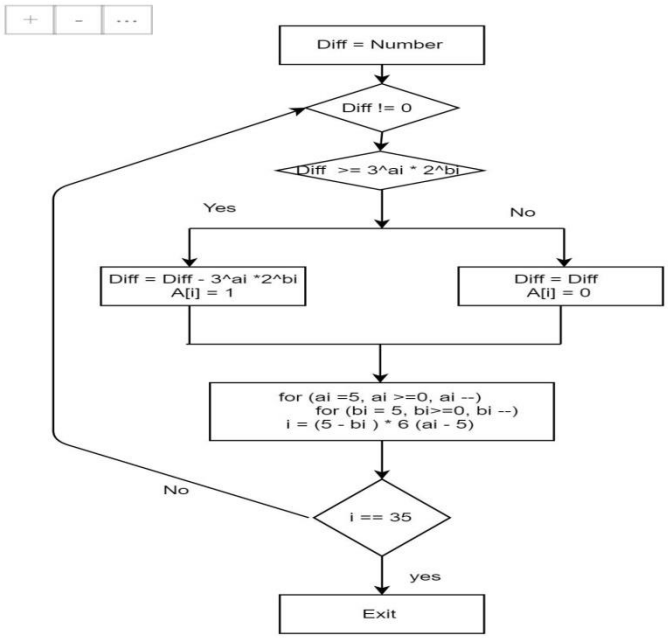
In this DBNS converter a decimal number can be represented as base of 3 and 2. Therefore 22 can be represented as $3^2\ 2^1 + 3^0\ 2^2$. Let us consider ai and bi as the powers of 3 and 2 $[3^{ai}\ 2^{bi}]$. The range of ai and bi can be varied depending of the number to be represented. For Ex. 8 bit number can be represented as powers of 3 and 2 with ai & bi, having range between 0 – 3. Number of combinations with this range is 3x3 = 9 bits. Therefore we will require 9 bit Register to represent DBNS number using binary number 0 & 1. Thus 22 can be represented as

	1	2	4
1	$0_{[8]}$	$0_{[7]}$	$1_{[6]}$
3	$0_{[5]}$	$0_{[4]}$	$0_{[3]}$
9	$0_{[2]}$	$1_{[1]}$	$0_{[0]}$

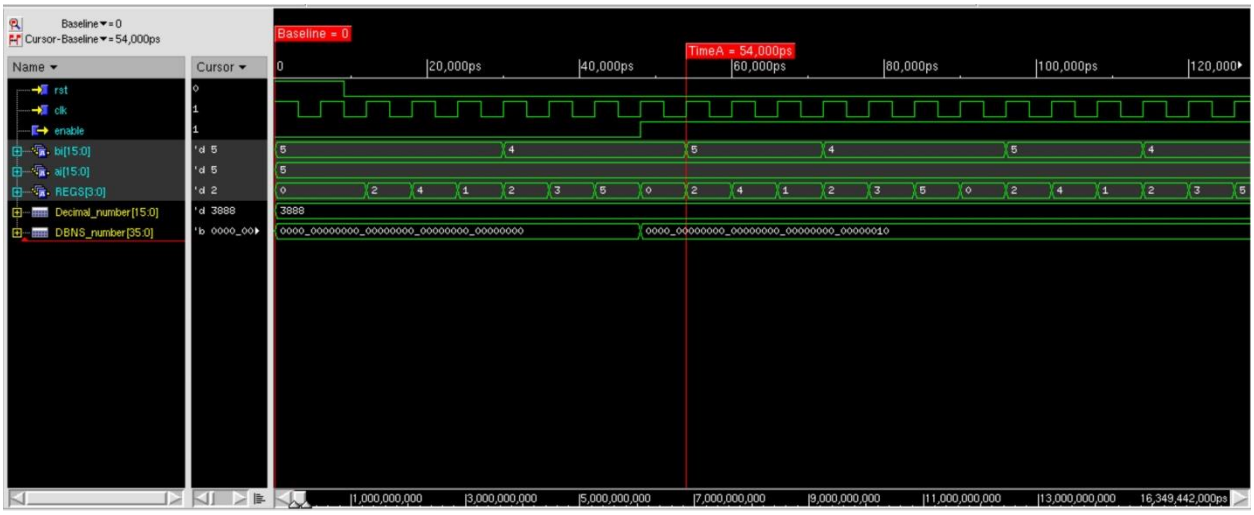
DBNS 22[9:0] = 0 0100 0010

Hence the following algorithm is used to represent a DBNS number. Store the value in a temporary variable called difference. Subtract difference this with next least value, which is combination of powers of 3 &2. The bit which represents this combination has to b set to 1. Perform this until difference is 0.

For Eg: diff = 22 - $3^2\ 2^1$ = 4 A[1] = 1 **->** diff = diff - $3^0\ 2^2$ A[6] = 1 diff = 0 **=>** A = 0 0100 0010



Time Take by the Converter(speed) = 45000ps

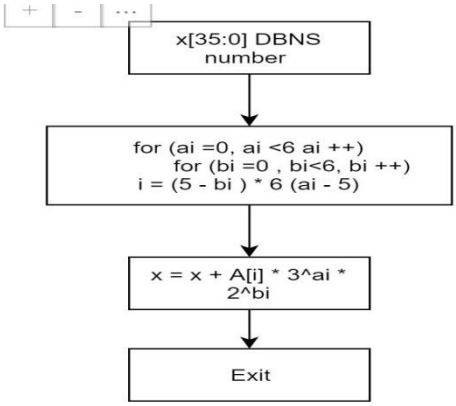


DBNS DeConverter

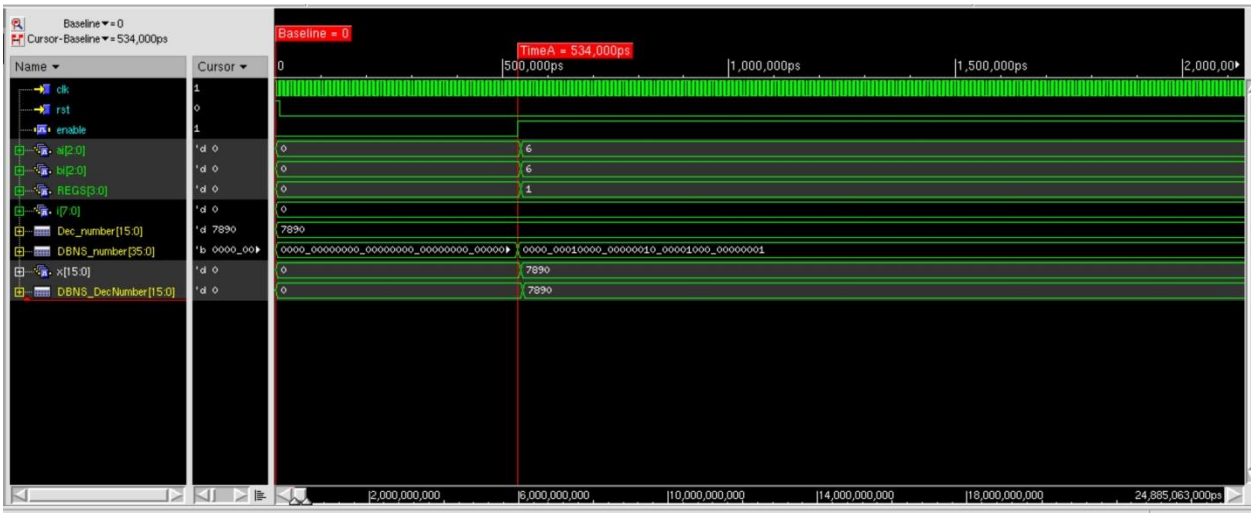
To convert DBNS number to a decimal number, we will have to multiply every bit of DBNS number with its respective exponent value and add it altogether.

For Eg: A = 0 0100 0010 x = x + A[1] * 3^2 2^1 = 18 x = 18 + A[6] * 3^0 2^2 x => 22

Below is the algorithm shown for DBNS Deconverter.



Time Take by the DeConverter(speed) = 525000ps



Part III

DBNS Multiplier

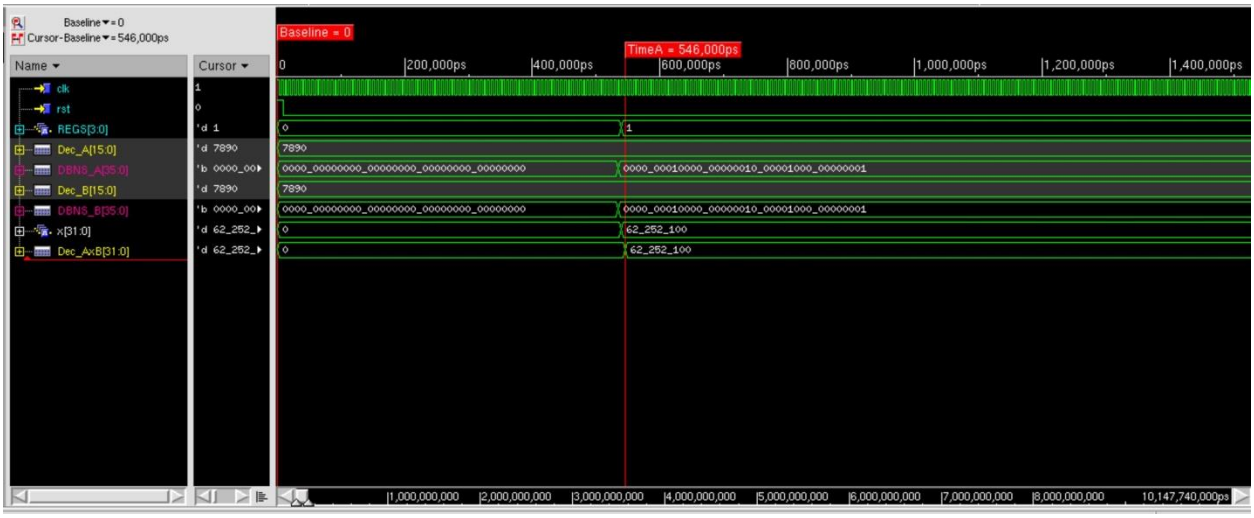
16 bit Decimal number received in REGA and REGB is sent to the converter and 35 bit A and B DBNS number along with an Enable signal. Once enable signal goes high, we perform DBNS multiplication i.e every exponential bit of A is multiplied with every exponential bit of B.

For Eg : 22 x 54 = (3^3 * 2^1) * (3^2 * 2^1) + (3^3 * 2^1) * (3^0 * 2^2)

= (3^5 * 2^2) + (3^3 * 2^3) = 1188

a. This Multiplier can be efficiently used by pipelining the converter and DEconverter operations such that when Multiplier is performing Multiplication of A0 & B0, Converter is converting A1 and B1 into DBNS number in parallel. Thus for a stream of inputs DBNS Multiplier can work efficiently.
b. Time Consumed to perform Multiplication is very less since operands are small numbers also if the bit value is 0 multiplication will not be performed. Hence number of multiplication operations performed p x m . Where p is the number of exponential components in number A and number of exponential components in number B.

Time Take by the DBNS_Multiplier(speed) = 546,000ps



Time Take by the DBNS_Multiplier without the converter or de converter i.e when the enable signal is 1(speed) = 12,054 ps

CODE:

Following zip file contains Test case and code for all the above :

Codes.zip